Cost-Eff ective Processes of Solar District Heating System Based on Optimal Artificial Neural Network

Mohamed Hany Abokersh,a Manel Vallès,a Laureano Jiménez,b Dieter Boer,a

aDepartament d’Enginyeria Mecànica, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain
bDepartament d’Enginyeria Química, Universitat Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain
Dieter.boer@urv.cat

Abstract

Aligning with the EU 2030 climate and energy package to achieve a share of at least 27% of renewable energies, and to improve the energy efficiency by at least 27%, the future solar district heating systems (SDHS) may enable the transition to a complete renewable society. Even though this promising tendency of the SDHS, a range of potential barriers are obstructing the wide deployment of SDHS and promoting high variation in quantifying the SDHS benefits over its lifetime. In this context, the optimization approaches are a viable option for determining the optimal structure, sizing, and operation of the SDHS. However, Meta-heuristics optimization models are computationally very expensive and have many limitations regarding the optimization process. Aligning with these challenges, this work tends to develop a robust Artificial Neural Network model based on Bayesian Optimization to solve the computational obstacle associated with heuristics optimization models for SDHS.

Keywords: Solar District Heating, Cost-effective, TRNSYS, Artificial Neural Network, Bayesian Optimization.

1. Introduction

One goal of the European Union “2030 Agenda for Sustainable Development” (European Commission, 2012) is the transition to a more efficient, and sustainable energy future that will include high shares of renewable energy in the global energy mix with major intention to cut 80% of the green emission. A promising pathway towards this vision lies in the adoption of solar district heating systems (SDHS). SDHS are placed in proximity to the energy end-use sector they serve, hence, minimizing energy transmission losses and incorporating locally available energy resources. Moreover, they typically incorporate multiple energy carriers and renewable and other efficient technologies that convert, store and deliver energy in the form of heating, cooling, and electricity or as other energy carriers (e.g., hydrogen) (Di Somma et al., 2015), which allows them to increase operational flexibility. Overall, SDHS delivers a series of economic, environmental, and technical benefits, as discussed in (Akorede et al., 2010). However, the vast quantity of combinations of available devices into a SDHS and the specific subsidies require a systematic analysis and evaluation method (Yang et al., 2015), making modeling and optimization approaches a viable option for determining the optimal structure, sizing and operation of SDHS.

During the design phase of a SDHS, several questions are answered by a quantitative analysis, which is usually performed with specific software simulation tools. However, the complexity of the system increases rapidly with the increment in the design variables. The coupling between the stochastic energy production and the energy consumption
determined by the system properties makes the influences of single elements hard to estimate without the appropriate tools. In this respect, software simulation tools (TRNSYS, Modelica, EnergyPlus, etc.) offer the possibility to achieve a high level of rationalization and transparency, enabling users to make informed choices. However, in order to produce accurate results, the level of abstraction of the system description must be accurately chosen. For instance, the system parametrization must cover all relevant variables, and the resolution of the weather data used must be relevant. Thus, a significant amount of time is required for completing a simulation with the use of a detailed (e.g., first principles) model, which we will call a fine model.

This computational obstacle of SDHS simulation may be overcome by supercomputers, cloud computing, or metamodeling. Supercomputers are expensive if not managed efficiently to avoid downtime. Cloud computing is presumably a cheaper alternative. Still, if the design team wishes to perform sensitivity analysis to identify relevant inputs or interaction effects, such analysis easily requires thousands of simulations to cover the design space sufficiently. This is likely to take hours or days – even with access to cloud computers (Yang, 2011). In this work we aim to create a fast-robust machine learning technique based on an artificial neural network (ANN) to predict the performance of the SDHS with diverse outputs. This will be a bridge to introduce the disciplines of Metaheuristics optimization in SDHS simulation models.

2. Methodology Framework

The robust machine learning framework of this study includes three sequential steps. First, a set of SDHS cases were created and simulated in TRNSYS with different decision variables for the sizing and operation of the equipment, resulting in a simulation generated database. The hyperparameters of the ANN, such as the number of hidden layers, the number of neurons per hidden layer, the activation function, etc. are often set via rules of thumb or trial and error. However, suitable hyperparameters depend on the system considered and generally cannot be determined beforehand. For this propose, the Bayesian optimization is employed to find a suitable set of hyperparameters for the prediction of the ANNs. Once the robust optimal design of the ANN is achieved, in the third step, the simulations generated data are used to train and validate the robust ANN-based model for predicting the performance of the SDHS instead of using a software simulation tool (TRNSYS).

2.1. SDHS Model in TRNSYS

The SDHS is designed to fulfill energy demands for space heating (SH) and domestic hot water (DHW) in a residential sector. Usually, these systems are designed to supply district heating for more than 100 apartments with a solar fraction of approximately 50%. The main components of the SDHS simulation model, as shown in Figure 1, are the thermal solar collector, the seasonal storage tank (SST), and the DHW storage tank (DHWT). The solar collector transfers the heat gained from the solar radiation to the storage tanks, which is then supplied to the customer on demand. The mismatching between the energy supply and demand in the daily and seasonal bases is balanced through the storage tanks. Auxiliary natural gas heaters are installed to back up the required heat demand in case the solar heating system failed to cover it. The SST based on water storage tank facilitates long-term storage of thermal energy used to cover the SH demand during the winter season with solar energy stored during a summer period. The long-term storage implies relatively large dimensions for the SST, which favors slow charging and discharging processes. On the other hand, the
DHWT is a short-term independent storage tank which is used to cover the daily DHW service at a temperature of 60°C. The proposed simulation model follows the models previously developed by Tulus et al. (Tulus et al., 2016).

Figure 1: Process flow diagram of the SDHS simulated in TRNSYS 18

2.2. Robust ANN model approach
The robust surrogate model building process has been divided into two main steps; the model setting selection and the model convergence testing. In the first step (Step A), this study emphasizes on the model settings and its relative hyperparameters through using the Bayesian optimization approach at different training set sizes. In the second step (Step B), the convergence of the ANN model is assessed through testing the develop a surrogate model based on the optimal model settings under a wide range of training set to define the optimal sample size.

3. Results and discussion
The results of the model setting and its convergence with various training sets are summarized in this section. First, the results of Bayesian optimization are analyzed for three primary training set to define the optimal setting model. Then, the ANN performance is tested under various training sets to identify the optimal sample size with consideration for the computational model expenses.

3.1. Model Setting
The solution of Bayesian optimization is given by an interactive parallel coordinate plot to identify suitable hyperparameters settings for the reduced number of configurations in step B (see. Figure 2). The plot shows the top 20% ranked optimal solutions results comprising 500 ANN model settings where each line represents one of these optimal solutions along with the achieved coefficient of variation (\(C.V\)) values. The table below the interactive parallel coordinate plots shows the optimal metamodel setting that achieves the highest accuracy at step (A) training sets. In case of no agreement for selecting a certain optimal setting at different sample sizes, the histogram attached to each interactive parallel coordinate column is utilized to propose the most frequently setting at each hyperparameter. The hyperparameters, including the training function, number of
layers, layer function, hidden function, and momentum mean at each training set, have the same optimal setting at different training sets, whereas the number of neurons and learning rate change at each training set. As observed from the histogram, most of the optimal results setting for the learning rate are set in a range below 0.01. Thus, the learning rate is set to 0.001 for the convergence stage (step B) to sustain the training set converge. On the other hand, the number of neurons with the size of 3, 14, and 20 are set for the convergence stage (Step B) since its optimal value is different for the training set size 64, 256, and 1024. A summary of the selected settings in the convergence stage (step B) is shown in the below table in Figure 2, where the nominated settings are highlighted.

<table>
<thead>
<tr>
<th>No. Samples</th>
<th>Train lr</th>
<th>No. Layers</th>
<th>No. Neurons</th>
<th>Layer func</th>
<th>Hidden func</th>
<th>LR</th>
<th>Mu</th>
<th>CV [100%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>64</td>
<td>3</td>
<td>3</td>
<td>Logist</td>
<td>Pneulin</td>
<td>0.042</td>
<td>0.004</td>
<td>49.7%</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>3</td>
<td>14</td>
<td>Logist</td>
<td>Pneulin</td>
<td>0.004</td>
<td>0.636</td>
<td>94.2%</td>
</tr>
<tr>
<td>1024</td>
<td>1024</td>
<td>3</td>
<td>20</td>
<td>Tanh</td>
<td>Pneulin</td>
<td>0.001</td>
<td>0.004</td>
<td>2.98%</td>
</tr>
</tbody>
</table>

Figure 2: interactive parallel coordinate plot combined with histograms are utilized to identify the optimal hypermeters setting for the ANN model in step A. The plot shows the top 20% ranked ANN model settings. The table below shows the optimal hyperparameters

Following the interactive parallel coordinate plot, a box plot (see. Figure 3) is built to show the performance of the three-training sets (64, 256, and 1024) based on the CV rank under the optimal selected hyperparameters with considering for three neurons sizes comprising 3, 14, and 20 in comparison to the default settings.

Figure 3: Box plot for the output in step A including the ANN model performance under the optimal and default settings
The box plot is characterized by the central mark and the upper and lower quartiles, which correspond to the box edge and show other optimal solutions. While the minimum and maximum optimal values are indicated at the whiskers. On the plot, the lined circles at each sample size show the results under optimal settings at a different number of neurons, whereas the cross symbols represent the results at the default settings. In general, the default setting does not yield to build accurate ANN models that approve the importance hyperparameters tuning. Moreover, fixing the number of neurons at 14 provides the most accurate results with a $C.V$ value of 24.1% and 9.2% for the 256 and 1024 training set, respectively.

3.2. Model Setting

In step B, we test the performance of the selected optimal hyperparameters at various training sizes in order to choose the most accurate ANN model with consideration for its efficiency in terms of the computational cost, as shown in Figure 4.

![Figure 4: Convergence of accuracy criteria at various training size with consideration for its relative computational cost](image)

In terms of the convergence, three accuracy criteria comprising the adjusted R-squared ($R^2$-adj), $C.V$, and Symmetric mean absolute percentage error (SMAPE) are utilized to evaluate the performance of the ANN model. The results show that the $R^2$-adj is a misleading criterion since most of the sample sizes exceed 97%. Therefore, using $C.V$ and SMAPE can be more efficient to measure the ANN model accuracy. Increasing the sample size has a clear tendency to improve the ANN model accuracy where the highest accurate value of 4.5% and 10% was indicated in a sample size of 2048 for the $C.V$ and SMAPE criterion, respectively.

In terms of the ANN model computational cost, an exponential behavior is indicated by increasing the sample size where the CPU time at 8192 sample size is $6\times10^4$ sec. Comprising the model accuracy with its efficiency simultaneously, the sample size 2048 provides the highest accuracy at an affordable computational time of $8.9\times10^3$ sec using an Intel® Xeon® E5-2620 v4 2.10 GHz processor with 32.0 GB RAM.

4. Conclusion

Following the high computational expenses obstacles associated with the SDHS simulation, this study proposes a complete framework based on a robust ANN model to
solve this computational obstacle. A summary of the robust ANN model key findings is the following:

- Relate to the ANN model settings, the hyperparameters comprising the number of hidden layers at 3, the number of neurons at 14, training function at Bayesian regularization, layer function at logsig, hidden function at purelin, learning rate at 0.001 and Momentum mean at 0.004 show the highest accurate ANN model at various training set size.

- Relate to the ANN model convergence at different training set sizes, the sample size of 2048 shows the highest accurate model prediction, where the $C.V$ criterion does not get below 4.5% for all model outputs at an affordable computational time of $8.9 \times 10^3$ sec.

In general, the proposed robust surrogate model built based on the two-model steps offers a sufficient approach for the construction of fast metamodels to overcome the computational barrier related to design space exploration, design optimization, and sensitivity analysis of heuristics optimization models.

Acknowledgment

The authors would like to acknowledge financial support from the Spanish Ministry of Economy and Competitiveness (RTI2018-093849-B-C33 (MCIU/AEI/FEDER, UE) and CTQ2016-77968 (MINECO/FEDER)) and to thank the Catalan Government for the quality accreditation given to their research group (2017 SGR 1409). This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713679 and from the Universitat Rovira i Virgili (URV).

References

https://doi.org/10.1016/j.rser.2009.10.025

https://doi.org/10.1016/j.enconman.2015.07.009


Klein, S.A. et al., 2004. TRNSYS Version. 18, Solar Energy Laboratory, University of Wisconsin-Madison, Website: <http://sel.me.wisc.edu/trnsys>.

https://doi.org/10.1016/j.apenergy.2016.08.037

https://doi.org/10.1016/j.envsoft.2010.10.007

https://doi.org/10.1016/j.energy.2015.03.101